



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/821,537	03/28/2001	Randall K. Curey	NG (NSD) 018807 US CIP	6434
26294 7590 12/22/2009 TAROLLI, SUNDHEIM, COVELL & TUMMINO L.L.P. 1300 EAST NINTH STREET, SUITE 1700 CLEVEVLAND, OH 44114				
EXAMINER				
LEE, PHILIP C				
ART UNIT		PAPER NUMBER		
2448				
MAIL DATE		DELIVERY MODE		
12/22/2009		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 09/821,537  
Filing Date: March 28, 2001  
Appellant(s): CUREY ET AL

---

Randall Curey  
Daniel Tazartes  
Kent Banno  
John Mark  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 10/29/09 appealing from the Office action mailed 08/03/09.

**(1) *Real Party in Interest***

The statement of the real party in interest in the brief is incorrect. The real party in interest is Litton Systems, Inc.

**(2) *Related Appeals and Interferences***

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) *Status of Claims***

The statement of the status of claims contained in the brief is correct.

**(4) *Status of Amendments After Final***

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) *Summary of Claimed Subject Matter***

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

4,109,311	Blum et al	08-1978
5,493,649	Slivka et al	02-1996
2005/0132375	Douceur et al	06-2005
5,621,663	Skagerling	04-1997
2006/0015719	Herbert et al	01-2006
6,223,201	Reznak	04-2001
5,826,092	Flannery	10-1998

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

1. This action is responsive to the amendment and remarks filed on April 1, 2009.
2. Claims 1-49 are presented for examination.

*Claim Rejections – 35 USC 102*

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-2, 4-9, 19-20, 22-27, 29-34, 44-45 and 47-49 are rejected under 35 U.S.C. 102(b) as being anticipated by Blum et al. (US 4,109,311), hereafter "Blum".

5. As to claim 1, Blum discloses the invention as claimed including a method for repetitively executing (col. 5, lines 35-42; and col. 6, lines 50-52) a plurality of software packages at one or more rates (the program requiring the greatest amount of processing time is allocated a greater number of time slices, col. 3, lines 47-54; fig. 3; col. 5, lines 24-42), utilizing a common set of computational resources (col. 3, lines 42-44), the method comprising: assigning a sequence of time intervals to each software package of the plurality of software packages (col. 1, lines 11-23; col. 3, lines 41-54; and col. 6, lines 2-9), the sequence of time intervals assigned to a particular software package of the plurality of software packages not overlapping the sequence of time intervals assigned to any other software package of the plurality of software packages (col. 3, lines 40-46); executing a subset of the plurality of software packages (abstract, lines 2-6; col. 2, lines 15-25), each respective software package in the subset plurality of

software packages being executed during predetermined time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages (abstract, lines 6-8; col. 1, lines 19-24; col. 3, lines 47-54; col. 6, lines 2-9; col. 7, lines 1-3).

6. As to claim 25, the claim is rejected for the same reasons as claim 1 above. In addition, Blum discloses an apparatus for practicing the method of claim 1 (Fig. 5).

7. As to claim 26, the claim is rejected for the same reasons as claim 1 above. In addition, Blum discloses an apparatus for repetitively executing (Fig. 5; and col. 5, lines 35-42; and col. 6, lines 50-52) a plurality of software packages at a plurality of rates (the program requiring the greatest amount of processing time is allocated a greater number of time slices, col. 3, lines 47-54; ; fig. 3; col. 5, lines 24-42), the apparatus comprising: a means for generating and assigning a sequence of time intervals to each software package of the plurality of software packages (col. 1, lines 11-23; col. 3, lines 41-54; and col. 6, lines 2-9), the sequence of time intervals assigned to a particular software package of the plurality of the plurality of software packages not overlapping the sequence of time intervals assigned to any other software package of the plurality of software packages (col. 3, lines 40-46); a means for executing a subset of the plurality of software packages (abstract, lines 2-6; col. 2, lines 15-25; and Fig. 5), each respective software package in the subset of the plurality of software packages is executed during predetermined time intervals defined by the sequence of time intervals assigned to the respective software package in the

subset of the plurality of software packages (abstract, lines 6-8; col. 1, lines 19-24; col. 3, lines 47-54; col. 6, lines 2-9; col. 7, lines 1-3).

8. As per claims 2 and 27, Blum further teach comprising the step: utilizing one or more tests to identify the plurality software packages that are valid (col. 4, lines 47-61), and wherein the subset of the plurality of software packages includes only valid software packages (col. 4, lines 47-61).

9. As per claims 4 and 29, Blum further teach wherein a given software package of the plurality of software packages is assigned a dedicated memory region, a given test of the one or more tests for validity being whether an address returned for an initialization procedure of the given software package of the plurality of software packages lies within the dedicated memory region of the given software package of the plurality of software packages (col. 4, lines 47-61).

10. As per claims 5 and 30, Blum further teach whether the address is returned for the initialization procedure of the given software package of the plurality of software packages within a predetermined time (col. 7, lines 1-10).

11. As per claims 6 and 31, Blum further teach wherein a given software package of the plurality of software packages is assigned a dedicated memory region, the dedicated memory region of the given software package of the plurality of software packages including a stack memory region and/or a heap memory region, a given test of the one or more tests for validity

being whether the stack memory region and/or the heap memory region assigned during the execution of an initialization procedure of the given software package of the plurality of software packages and various associated entry points lies within the dedicated memory region assigned to the given software package of the plurality of software packages (col. 4, lines 47-61).

12. As per claims 7 and 32, Blum further teach whether the stack memory region and/or the heap memory region and the various associated entry points are returned within a predetermined time (col. 7, lines 1-10).

13. As to claims 8 and 33, Blum discloses a given software package of the plurality of software packages is assigned a dedicated memory region (fig. 3; col. 3, line 65 to col. 4, line 13; and col. 4, lines 50-54).

14. As per claims 9 and 34, Blum further teach the dedicated memory region assigned to the given software package of the plurality of software packages includes a stack memory region in which a stack of the given software package of the plurality of software packages resides (30, 33, fig. 2; col. 4, lines 14-46).

15. As per claims 19 and 44, Blum further teach each software package of the plurality of software packages is assigned a memory block (30,33, fig. 2), a given software package of the plurality of software packages being configured to read data only from zero or more memory blocks associated with other software packages of the plurality of software packages, the zero or

more memory blocks readable by the given software package of the plurality of software packages being either predetermined or determined during execution of the given software package of the plurality of software packages in accordance with a set of one or more rules (col. 4, lines 14-46).

16. As per claims 20 and 45, Blum further teach each software package of the plurality of software packages is assigned a memory block (30,33, fig. 2), a given software package of the plurality of software packages being configured to write data only to zero or more memory blocks associated with other software packages of the plurality of software packages, the zero or more memory blocks writeable by the given software package of the plurality of software packages being either predetermined or determined during execution of the given software package of the plurality of software packages in accordance with a set of one or more rules (col. 4, lines 14-46).

17. As per claims 22 and 47, Blum further teach a presence of subset of the plurality of software packages of the plurality of software packages is detected (col. 4, lines 34-61).

18. As to claims 23 and 48, Blum discloses one or more software packages is independently compiled, linked, and loaded (col. 3, line 65 to col. 4, line 13; and col. 4, lines 50-54) (i.e., programs are transfer to control storage (loading) and executed (col. 4, lines 4-33) (compiling). Blum et al further teach pointer linking a program to be executed (col. 4, lines 47-61) (linking)).

19. As per claims 24 and 49, Blum further teach each software package of the plurality of software packages has a stack, that is selected prior to executing the software package (30, 33, fig. 2; col. 4, lines 14-46).

*Claim Rejections – 35 USC 103*

20. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

21. Claims 3 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blum in view of Slivka et al, U.S. Patent 5,493,649 (hereinafter Slivka).

22. As per claims 3 and 28, Blum teaches the invention as claimed in claims 2 and 27 above. Blum does not teach one's complement checksum test. Slivka teaches a given test of the one of the tests for validity is a one's complement checksum test of a software package's program memory data (col. 1, lines 56-67; col. 4, line 51-col. 5, line 2).

23. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Slivka because Slivka's teaching of one's complement checksum test would allow Blum's system to validate a program when multiple programs are executing within the memory.

24. Claims 10-11 and 35-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blum in view of Douceur et al, U.S. Patent Application Publication 2005/0132375 (hereinafter Douceur).

25. As per claims 10 and 35, Blum teaches the invention as claimed in claims 1 and 26 above. Blum does not teach background tasks as well as foreground tasks. Douceur teaches background tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed ([0005]).

26. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Douceur because Douceur's teaching of background tasks and foreground tasks would increase the efficiency of Blum's system by allowing a background process to be operated when no other priority process is using the processor.

27. As per claims 11 and 36, Blum and Douceur teach the invention substantially as claimed in claims 10 and 35 above. Douceur further teaches a background task is an infinite loop ([0048]).

28. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Douceur for the same reason as claim 10 above.

29. Claims 13-17 and 38-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blum in view of Skagerling, U.S. Patent 5,621,663 (hereinafter Skagerling).

30. As per claims 13 and 38, Blum teaches the invention as claimed in claims 1 and 26 above. Blum does not teach a failure log. Skagerling teaches a failure in the execution of a given software package causes information to be logged in a failure log (col. 4, lines 54-57).

31. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Skagerling because Skagerling's teaching of failure log would increase the alertness of Blum's system by allowing logged failure to be reported and analyzed.

32. As per claims 14 and 39, Blum and Skagerling teach the invention substantially as claimed in claims 13 and 38 above. Skagerling further teach a failure in execution is linked to the given software package that caused the failure (col. 2, line 51-col. 3, line 6).

33. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Skagerling for the same reason as claim 13 above.

34. As per claims 15 and 40, Blum and Skagerling teach the invention substantially as claimed in claims 13 and 38 above. Skagerling further teach wherein quality of performance in executing the given software package is represented by one or more performance-quality parameters (col. 3, lines 57-63), values of the one or more performance-quality parameters being determined from the information logged in the failure log, the execution of the given software package being subject to a plurality of execution options, an execution option being selected on the basis of the values of the one or more performance-quality parameters (col. 5, line 64-col. 6, line 6).

35. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Skagerling for the same reason as claim 13 above.

36. As per claims 16 and 41, Blum and Skagerling teach the invention substantially as claimed in claims 15 and 40 above. Skagerling further teach the plurality of execution options are user configurable (col. 4, lines 37-41).

37. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Skagerling for the same reason as claim 13 above.

38. As per claims 17 and 42, Blum and Skagerling teach the invention substantially as claimed in claims 15 and 40 above. Skagerling further teach wherein the performance-quality parameters include the number of failures and/or the rate of failures for one or more classes of failures recorded in the failure log (col. 5, line 57-col. 6, line 6).

39. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Skagerling for the same reason as claim 13 above.

40. Claims 18 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blum in view of Herbert et al, U.S. Patent Application Publication 2006/0015719 (hereinafter Herbert).

41. As per claims 18 and 43, Blum teaches the invention as claimed in claims 1 and 26 above. Blum does not teach safety-critical software. Herbert teaches safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software ([0021], [0025], [0041]).

42. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Herbert because Herbert's teaching of safety-critical software would improve the reliability of execution in Blum's system by allowing system critical software to be executed in isolation in order avoid interface caused by other non-critical software failure.

43. Claims 21 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blum in view of Reznak, U.S. Patent 6,223,201 (hereinafter Reznak).

44. As per claims 21 and 46, Blum teaches the invention as claimed in claims 1 and 26 above. Although Blum teaches an executive software package enforces the discipline that each of the one or more software packages of the plurality of software packages being executed software package executes only during the time intervals of its sequence of time intervals (col. 1, lines 17-60), however, Blum does not teach when the execution of a software package extends into a time interval assigned to another software package. Reznak teaches the executive software package determining when the execution of the respective software packages of the plurality in the subset of the software packages is executed a software package extends into a time interval defined by the sequence of time intervals assigned to at least one different software package in the subset of the plurality of software packages and performs a remedial action (col. 5, lines 10-15; col. 6, lines 24-32).

45. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum and Reznak because Reznak's teaching of determining when the execution of any one of the one or more software packages of the plurality of software packages being executed a software package extends into a time interval belonging to the sequence of time intervals assigned to another of the one or more software packages would increase the flexibility of Blum's system by allowing adjustment of the assigned time intervals for execution by the programs.

46. Claims 12 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blum and Douceur in view of Flannery, U.S. Patent 5,826,092 (hereinafter Flannery).

47. As per claims 12 and 37, Blum and Douceur teach the invention substantially as claimed in claims 10 and 35 above. Blum and Douceur do not teach minimize the power utilized. Flannery teaches causing the power utilized in executing the given software package to be minimized after completion of the background tasks (col. 3, lines 51-58).

48. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Blum, Douceur and Flannery because Flannery's teaching of minimize the power utilized would increase the efficiency of Blum's and Douceur's systems by allowing their system to conserve more power.

The examiner summarizes the various points raised by the appellant and addresses replies individually.

Appellant argued that:

- (1) Blum fails to teach a sequence of time intervals are assigned to each software package of a plurality of software packages, and that a subset of the plurality of software packages are executed during predetermined time intervals defined by the assigned sequences of time.
- (2) The combination of Blum and Reznak in rejecting claims 21 and 26 is not obvious.

**In reply** to argument (1): on page 7, lines 10-15 of the appeal brief filed on 10/29/09, appellant states "Blum does not disclose that multiple sequences of time are assigned to each software package of a plurality of software packages, and that a subset of the plurality of software packages are executed during predetermined time intervals defined by the assigned sequences of time as recited in claims 1 and 26." It is noted that the features upon which applicant relies (i.e., multiple sequences...) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). On page 8, lines 18-25 of the appeal brief, appellant further states "Appellant's representative respectfully submits that the Examiner's interpretation of Blum appears to be contrary to the decision issued by the Board of Patent Appeals and Interferences (BPAI) issued on September 18, 2008 for the present Application. Specifically, page 19 of the BPAI decision states that Blum

discloses generating a recurring set of time slice intervals and assigning a time slice to a particular program (emphasis added). The BPAI decision also states that (at the time of the Appeal) in the claims, there is no requirement that a particular sequence of time be assigned to a particular software package." Although page 19 of the BPAI decision does states "there is no requirement that a particular sequence of time be assigned to a particular software package", however, the BPAI does not state that Blum fails to teach "assigning a sequence of time intervals to each software package of a plurality of software packages." Blum does teach a sequence of time intervals are assigned to each software package of a plurality of software packages, and that a subset of the plurality of software packages are executed during predetermined time intervals defined by the assigned sequences of time. Specifically, Blum teaches a plurality of software programs (e.g., programs 0, 1 and 2 in figs. 2 and 3). Blum further teaches each program is sequentially allocated time slices for the duration of which all the resources of the processing unit are made available to one program (col. 3, lines 41-54). For example, Blum teaches a time slice cycle is subdivided into six time slice intervals and the six time slice intervals are divided equally among three programs. Thus, each program is serviced twice during the assumed six time slice cycle (col. 6, lines 43-52). This means that Blum teaches assigning (i.e., allocating) a sequence of time intervals (i.e., arrangement of time slice intervals, e.g., two of the six time slice intervals in the time slice cycle) to each software package of a plurality of software packages (i.e., to each program of a plurality of programs) and that a subset of the plurality of software package (i.e., one or more of the individual program of the three programs) are executed during predetermined time intervals defined by the assigned sequence of time (i.e., Serviced by the processing unit during the predetermined time slice assigned to each program, e.g., serviced

twice (i.e., assigned sequence of time) during the six time slice intervals (i.e., predetermined total time slices)). It is noted that the examiner interprets the term “sequence” based on The American Heritage Dictionary as an arrangement. Therefore, “sequence of time” is interpreted as “arrangement of time”.

**In reply** to argument (2): on page 12, lines 9-12 of the appeal brief, appellant states “Since in Blum, programs are selected to be executed in real time, there is never a time that a program is being executed in a time slice that is associated with different program”. Examiner disagreed. Blum teaches dynamically changing the time slice assignment. Blum teaches the individual programs can automatically add to their share of computing time (col. 5, lines 18-19). A program can add to its computing time by taken from the computing time (e.g., time slices) of other programs (col. 5, lines 18-23; col. 12, lines 28-31). This means a program in Blum’s system can execute in the computing time that is associated with different program. Therefore, it would be obvious to combine Blum’s system and Reznak’s teaching because Reznak’s teaching of determining when the execution of any one of the one or more software packages of the plurality of software packages being executed a software package extends into a time interval belonging to the sequence of time intervals assigned to another of the one or more software packages would increase the flexibility of Blum’s system by allowing adjustment of the assigned time intervals for execution by the programs. Furthermore, Blum teaches scheduling of time intervals for execution including dynamically determining/changing the assignment of processing time. Reznak teaches scheduling of time intervals for execution including determining assignment of time is exceeded. Therefore, Blum and Reznak are analogous arts that would be obvious to combine.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

**(12) Conclusion**

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Philip Lee

December 17, 2009

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2451

Conferees:

/FIRMIN BACKER/

Supervisory Patent Examiner, Art Unit 2448

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2451